

FindBugs

Sang Shin
www.jPassion.com
“Learn with jPassion!”



Topics

- What is & Why FindBugs?
- Bug categories
- How to get started?
- Integration with other tools

What is & Why FindBugs?

What is FindBugs?

- FindBugs looks for bugs in Java programs
 - > It uses static analysis to inspect Java bytecode for occurrences of bug patterns - executing the program is not necessary, source code is not even needed
 - > It is not a style checker
- It is based on the concept of bug patterns. Bug patterns arise for a variety of reasons:
 - > Difficult language features
 - > Misunderstood API methods
 - > Misunderstood invariants when code is modified during maintenance
 - > Garden variety mistakes: typos, use of the wrong boolean operator

Bug Categories

Bug Categories

- Bad practice
- Correctness
- Internationalization
- Malicious code vulnerability
- Multithreaded correctness
- Performance
- Security
- Dodgy code

<http://findbugs.sourceforge.net/bugDescriptions.html>

Examples: Bad Practices

- Dm: Method invokes `System.exit(...)` (DM_EXIT)
 - > Invoking `System.exit` shuts down the entire Java virtual machine. This should only be done when it is appropriate.
 - > Such calls make it hard or impossible for your code to be invoked by other code. Consider throwing a `RuntimeException` instead.
- DMI: Don't use `removeAll` to clear a collection (DMI_USING_REMOVEALL_TO_CLEAR_COLLECTION)
 - > If you want to remove all elements from a collection `c`, use `c.clear`, not `c.removeAll(c)`.
 - > Calling `c.removeAll(c)` to clear a collection is less clear, susceptible to errors from typos, less efficient and for some collections, might throw a `ConcurrentModificationException`.

Examples: Correctness

- BC: Impossible cast (BC_IMPOSSIBLE_CAST)
 - > This cast will always throw a ClassCastException. FindBugs tracks type information from instanceof checks, and also uses more precise information about the types of values returned from methods and loaded from fields.
 - > Thus, it may have more precise information than just the declared type of a variable, and can use this to determine that a cast will always throw an exception at runtime.
- BC: instanceof will always return false (BC_IMPOSSIBLE_INSTANCEOF)
 - > This instanceof test will always return false. Although this is safe, make sure it isn't an indication of some misunderstanding or some other logic error.

Examples: Internationalization

- Dm: Reliance on default encoding (DM_DEFAULT_ENCODING)
 - > Found a call to a method which will perform a byte to String (or String to byte) conversion, and will assume that the default platform encoding is suitable. This will cause the application behaviour to vary between platforms. Use an alternative API and specify a charset name or Charset object explicitly.

How to Get Started

Steps for Getting Started

- Running FindBugs
 - > `java -jar $FINDBUGS_HOME/lib/findbugs.jar options`
- Options
 - > `-gui`: runs the graphical user interface (GUI)
 - > `-textui`: runs the command line user interface
 - > `-version`: displays the FindBugs version number
 - > `-help`: displays help information for the FindBugs command line user interface

Integration with other Tools

Integration with other tools

- Ant
- Maven
- IDE
 - > Eclipse, NetBeans, IntelliJ IDEA, etc
- Continuous Integration Tool
 - > Hudson, Jenkins
- Continuous Inspection Tool
 - > Sonar

Thank you!

Check JavaPassion.com Codecamps!
<http://www.javapassion.com/codecamps>
“Learn with Passion!”

